# Critique of DELFIC's Cloud Rise Module

Vincent J. Jodoin, Capt., USAF

Doctoral Candidate - Nuclear Engineering

AFIT DS94J ENP



DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/EN/TR/93-04

# Critique of DELFIC's Cloud Rise Module

Vincent J. Jodoin, Capt., USAF

Doctoral Candidate - Nuclear Engineering
AFIT/DS94J/ENP

# Abstract

This report discusses the Cloud Rise Module (CRM) of the Defense Land Fallout Interpretive Code (DELFIC), the DOD's reference fallout code. The first section discusses the errors found in the 1979 documentation. These errors have been corrected and further research by the Air Force Institute of Technology will include the corrections listed in this section. The second section will discuss the use of Mathematica in simulating the CRM. This part of the report will describe how a higher level language was used to review the results of DELFIC's CRM and uncover some problem areas. The final part of this report will describe further work as planned by the author in improving the CRM of DELFIC.

# Table of Contents

# List of Figures

# Critique of DELFIC's Cloud Rise Module

## 1.0 Introduction

This report is a critique of the Defense Land Fallout Interpretive Code (DELFIC) as it pertains to the author's research into nuclear cloud rise and growth. The version of DELFIC being reviewed in this report is the last well documented change which gave reasonable results (Norment, 1979a & 1979b). Other efforts have gone into DELFIC since then but have either not been well documented or give questionable results. The 1979 version was recovered by McGahan with SAIC and provided to AFIT for this research. Throughout this report, "the documentation" refers to Volume I (Norment, 1979a) and Volume II (Norment, 1979b) of the 1979 version of DELFIC.

The current AFIT research is focused on the Cloud Rise Module (CRM) of DELFIC. The CRM includes the main subprogram ICRMEX and its subordinate routines, which determine the initial conditions, solve the cloud rise equations and prepare the definition of the particles aloft for the diffusive transport module (DTM). The code in the CRM was compared to the listings in Volume II.

This report has three parts. The first section discusses the errors found in the documentation. These errors have been corrected and further research by AFIT will include the corrections listed in this section. The second section will discuss the use of Mathematica in simulating the CRM. This part of the report will describe how a higher level language was used to review the results of DELFIC's CRM and uncover some problem areas. The final part of this report will describe further work as planned by the author in improving the CRM of DELFIC.

## 2.0 Corrections Needed in the 1979 DELFIC CRM

The last well documented version of DELFIC is Norment's 1979 version. In studying this version, a few errors were found both in the volume pertaining to the theory and the volume which included the source code. Below is a discussion of those discrepancies and the corrections needed to remedy them. When reviewing these errors, it may be beneficial to have a copy of Norment's two volumes from 1979 on hand.

## 2.1 Changes needed to compile CRM

In recovering the 1979 version of DELFIC, McGahan had to add a MAIN program, since the documentation was lacking one. Since AFIT's research is focused on the CRM of DELFIC, calls in the MAIN program to the DTM or the output processor module (OPM) were commented out. The CRM compiled with no changes on a 386-SX-16 using Microsoft FORTRAN 5.1. To compile on a SPARC station 2 using SUN FORTRAN 1.4, two separate FORMAT statements (016 and 099) in the DBG subroutine, had to be edited. The line numbers as listed in Volume II were DBG 25-DBG 27 and DBG 30-DBG 34. To prevent ambiguity in the statements, commas were needed after the X's, which represent spaces in FORTRAN output. These changes allowed the CRM to be compiled without any warnings or errors.

Similar changes were needed to compile the entire DELFIC program but will not be listed here. The documentation contains a test case's input and output which was used to verify that the code was running correctly. After the CRM was compiled successfully, the program's output was verified with the output as listed in Volume II. Additional post-processors were built with Mathematica to display the CRM variables as a function of time. The code is fast on today's computers with the entire DELFIC test case taking only 20 seconds on a 486-33.

## 2.2 Errors in 1979 Volume I

There were two errors in Volume I. Both pertained to the definition of the symbol for the ratio of molecular weights of water to air, $\xi$. The errors are described in the following sections with the heading giving the page number in Volume I.

### 2.2.1 Definition of $\xi$ pages 14 and 93

$\xi$ is incorrectly defined as the ratio of molecular weights of air to water, 29/18. In fact, after checking with earlier documentation as well as other references, the constant $\xi$ is that of <u>water over air</u>, 18/29. This is only an error in Volume I and therefore the code has the correct definition for $\xi$.

### 2.2.2 Definition of $x_e$ page 13

The above error affects the definition of $x_e$, Eq. (2.1.11).

$$x_e = \frac{H_R P_{ws}}{\xi P} \quad \text{(EQ 1)}$$

The $\xi$ in this equation should be in the numerator not the denominator. This is the only occurrence of misusing $\xi$ since this is the only new equation in the 1979 version that uses $\xi$. All the other equations which use $\xi$ were derived in earlier versions (Heubsch, 1967; Norment, 1970; Norment, 1977) with its proper definition. These other equations are repeated in their correct form in the 1979 version.

This error was found using Mathematica. By coding the equations as they were presented in Volume I, it was noticed that the value for $x_e$ was a factor $\xi^2$ too low. This caused more detailed scrutiny of the equation until the error was found.

## 2.3 Errors in 1979 Volume II

The errors in Volume II have to do with the code listings for the CRM. The errors are described in the following sections with the heading giving the line number of the subroutine which contains the error along with the page number in Volume II. All corrections were made individually to check the effect on the 1979 test case (a 50 kt, 0 ft. HOB shot). The results are shown along with the original, uncorrected results (see Figure ORIGINAL). *It should be noted that even when the effect on this test case is negligible, the effect of the error may be greater with other scenarios.*

### 2.3.1 CRMIN 66 page 78

```
7 soilht=ssam*(tad+781.6*(tpr-te)+0.2856*(tpr**2-te**2)+
```

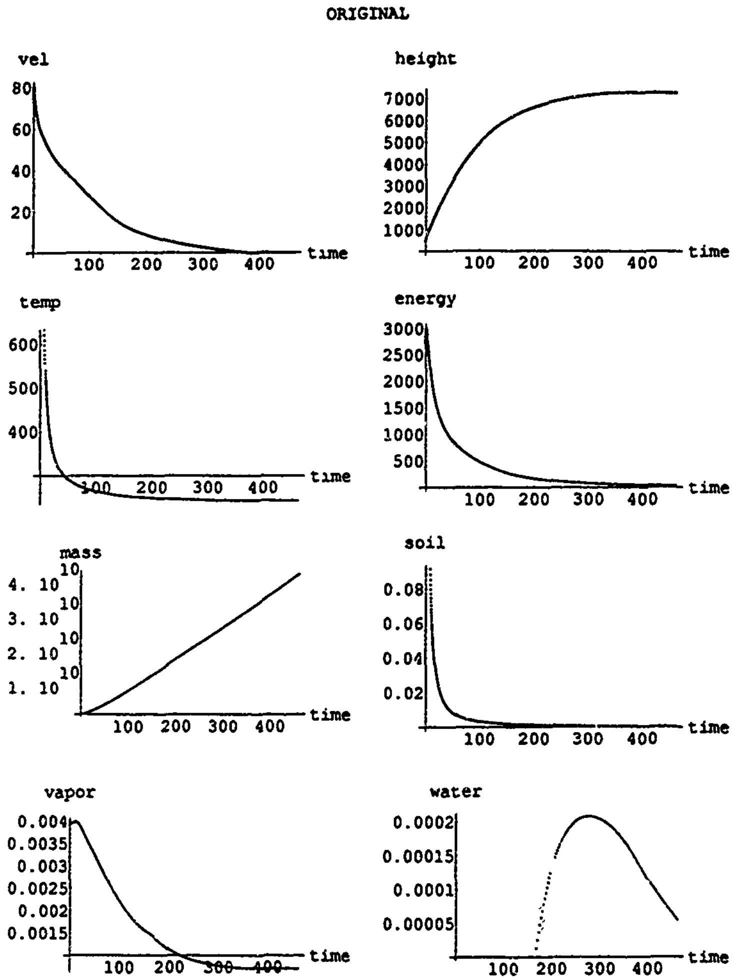
The documentation gives the definitions for the specific heat of soil, $c_s$, on page 13 of Volume I. When the code does the integration of the definition for $c_s$, the term 0.5612 T should go to 0.2806 $T^2$. Instead the code lists it as 0.2856 as shown above. All that needs to be done is to change the 5 into a 0. This has little effect on the test case.

### 2.3.2 DERIV 91 page 85

```
qq=qt*qx*qxe*(1.+x+wt)/(1.+x+s+wt)
```

In the code $\beta'$ is incorrectly calculated (1.+X+WT)/(1.+X+S+WT) as shown above. The numerator should be (1.+X), consistent with the true definition for $\beta'$(see page 92 Volume

**FIGURE 1. Original uncorrected CRM variables (MKS units)**



I). The CRM uses this variable in defining the first term on the right hand side of Eq. (2.2.4) on page 21 of Volume I.

$$\frac{dE}{dt} = 2k_2 \frac{T^*}{T_e^*} \beta' \frac{u^2 v}{H_c} + \frac{u^2}{2} \frac{1}{m} \frac{dm}{dt}\bigg|_{ent} - E \frac{1}{m} \frac{dm}{dt}\bigg|_{ent} - \varepsilon \qquad \text{(EQ 2)}$$

This change had little effect on the test case.

### 2.3.3 DERIV113 page 86

```
100 drm=(rm/(1.-cpai/(cp*t*qx)))*rmix*(rs *rl+(qt*qx*qxe*9.8*u-eps)*
```

This line of code is used to calculate the rate of mass entrained as defined in Eq. (2.2.5D) on page 22 of Volume I.

$$\frac{dm}{dt}\bigg|_{ent} = \frac{\beta' m}{1 - \frac{\beta'}{T^* \bar{c}_p} \int_{T_e}^{T} c_{pa}(T)\, dT} \left( \frac{S}{V} \mu v + \frac{\beta'}{T^* \bar{c}_p} \left[ \frac{T^*}{T_e^*} g u - \varepsilon \right] - \frac{g u}{R_a T_e^*} \right) \qquad \text{(EQ 3)}$$

The error occurs in the fraction before the integral sign. The code as listed above deletes the $\beta'$ and uses $c_p$ instead of $\bar{c}_p$. To remedy this, the line was replaced with the following line, noting that it is now longer than the 72 characters allowed in FORTRAN, requiring the excess characters to be placed on line DERIV114.

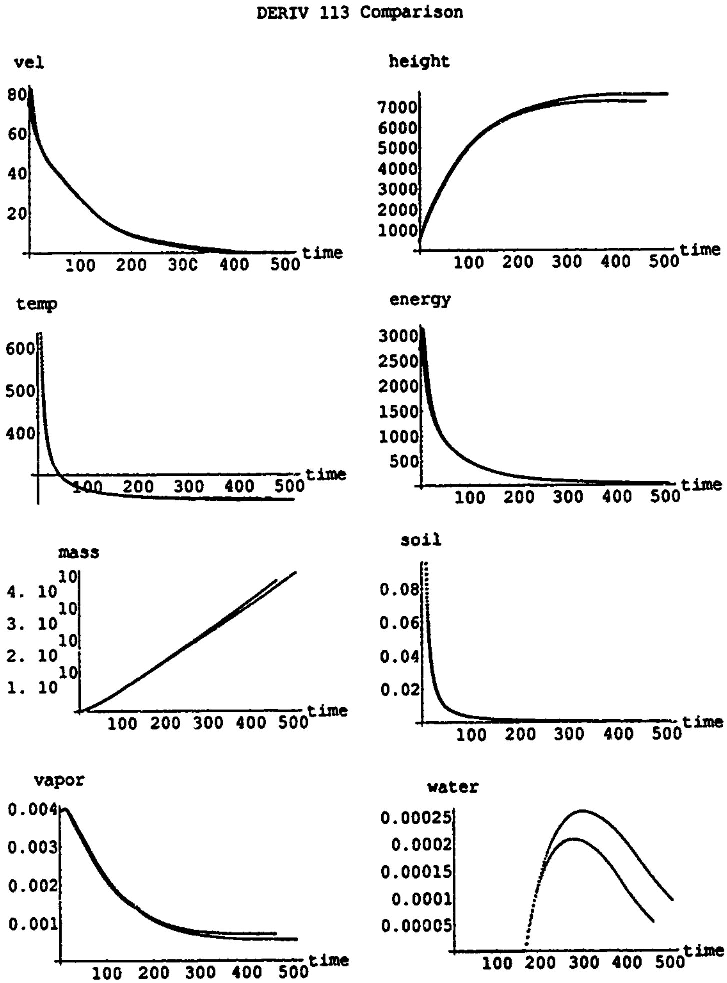
```
100 drm=(rm/(1.-rmix*cpai/(cr*t*qx)))*rmix*(rs *rl+(qt*qx*qxe*9.8*u-eps)*
```

This change has a small but noticeable effect on the test case (see figure DERIV 113 Comparison).

### 2.3.4 ATMR 187 thru ATMR 203 page 72

```
        do 260 i=2,256
        alt(i)=alt(i-1)+dalt
225     if(a1.ge.alt(i))go to 250
        if(alt(i)-a1 .lt. 2.) go to 250
        na=na+1
        if(nat-na .ge.0)go to 240
230     irror=-230
        go to 130
240     read(irise)a1,a2,a3,a4,a5,a6
        go to 225
250     terp= dalt /(a1-alt(i-1))
        atp(i)=atp(i-1)+terp*(a2-atp(i-1))
        prs(i)=prs(i-1)+terp*(a3-prs(i-1))
        rlh(i)=rlh(i-1)+terp*(a4-rlh(i-1))
```

**FIGURE 2. Effect of error in entrainment equation, DERIV 113 Comparison**



```
          rho(i)=rho(i-1)+terp*(a5-rho(i-1))
          eta(i)=eta(i-1)+terp*(a6-eta(i-1))
  260     continue
```

The above loop expands the atmosphere tables to 256 entries from -1,000m to 50,000m in 200m increments. It is supposed to be using linear interpolation of the input data to do this, but it isn't. In reality it is linear interpolating between the last 200m increment value it has calculated and stored and the next input datum. This causes a skewing of the data in the table, which is used quite extensively throughout the CRM.

The best way to correct this is to not make an artificial regular spaced table to interpolate from, but rather, linear interpolate on the input data itself when needed. This has a small effect on the test case.

### 2.3.5 DERIV 54 thru DERIV 61 page 85

```
          rmix=(1.+x)/(1.+x+s+wt)
          cr=cp*rmix
          if(tmps-t)380,381,381
  381     if(t-848.)3810,3810,3811
  3810    cs=781.6+0.5612*t-1.881e7/t**2
          go to 3812
  3811    cs=1003.8+0.13510*t
  3812    cr=cr+cs*(s+wt)/(1.+x+s+wt)
```

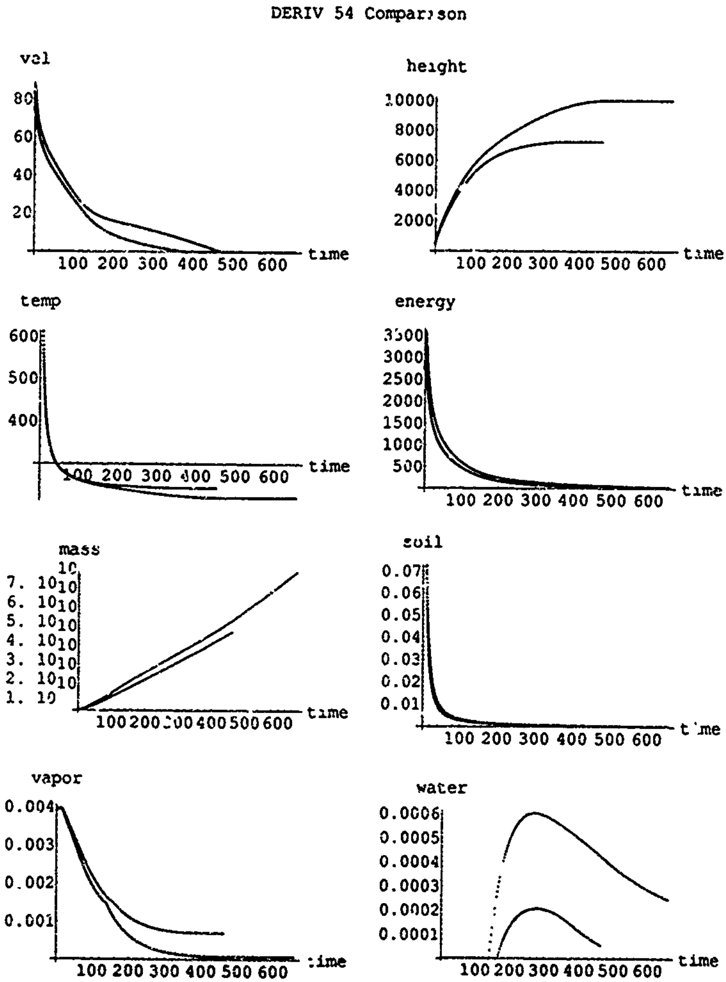
The definition for $\bar{c}_p$ at the top of page 21 Volume I doesn't quite match the lines from Volume II above. The above code actually has a switch in the definition of $\bar{c}_p$ which is not mentioned in Volume I.

$$\bar{c}_p = \beta' c_p + (1-\beta') c_s k(T, T_s) \qquad \text{(EQ 4)}$$

If the temperature of the cloud is greater than the initial temperature of the soil in the cloud, $k(T, T_s)=0$. When the cloud temperature drops below the initial soil temperature, $k(T, T_s)=1$. This is equivalent to stating that the heat content of the soil is negligible compared to that of the air mixture until the cloud has cooled to the initial soil temperature.

If one models the equations as described in Volume I, always accounting for the soil heat, a noticeable difference is seen in the behavior of the cloud (see Figure DERIV 54 Comparison). This is exactly how this difference was found, by modeling the Volume I equations, as is, in Mathematica. The difference seen using just the test case from 1979 shows that ignoring the heat content of the soil is not a very good approximation.

FIGURE 3. Effect of error in definition for $\hat{c}_p$, DERIV 54 Comparison



# 3.0 Lessons Learned Using Mathematica

Mathematica is a higher level programming language with symbolic, numeric, and graphic capabilities. By combining all three formats in one package, along with hundreds of built in functions/routines, this type of language can be used as a separate debugging tool when verifying the workings of a large nested, conventional program. It was especially helpful since DELFIC was written by many different authors using different versions of FORTRAN. Rather than trying to follow someone else's programming from start to finish, a higher level language can be used to model the theory and check for accuracy.

When taking such an approach, errors may be found a little easier since whole sections of conventional codes are replaced by a few lines of higher level code or one built in function This approach also provides a way of seeing better ways of solving a problem. This section describes some of these findings not previously mentioned in the above chapter.

## 3.1 Method of solving the set of cloud equations

One area of DELFIC in which the higher level language provided some insight was the method of solving the system of coupled cloud equations (see Appendix A). The set of eight first order nonlinear differential equations are coupled, so that some equations use the derivative of another variable. DELFIC uses a fourth order Runge-Kutta-Gill method with a fixed time step for each of three time domains. For the first second, the equations use a1/32 second time step, then a 1/4 second time step until 100 seconds, and finally a 2.5 second time step until stabilization. When transitioning from no condensed water to condensed water present, a 1/4 second time step is used to better define when this transition takes place.

The built in function for Mathematica which solves systems of ODEs uses a variable time step and adapts to a different method depending on the stiffness of the equations. Because the equations in DELFIC are stiff, the Runge-Kutta-Gill method may not be a good choice for solving them. This may be one area where DELFIC needs improvement. The question also comes up as to whether or not DELFIC uses the proper time steps for numerical accuracy, and if so, is it the best choice for computational efficiency.

## 3.2 Transition from dry to wet equations

As mentioned above, DELFIC finds where the onset of condensation occurs in the cloud using a 2.5 second time step, but then backs up one step and finds the onset more precisely.

It does this by going back to a 1/4 second time step until the transition and returning to a 2.5 second time step once its found. This was simulated with the higher level language, but a discrepancy still existed between the output of the two different languages.

It was determined that the difference was due to the fact that DELFIC fixes the equations as either wet or dry for the entire time step. Since its a four step method for each time step, the equations should be allowed to switch as necessary *within* the time step. This is precisely what Mathematica does. DELFIC may be trying to compensate for the fixing of the dry/wet mode during a four step method with its retrace using a smaller time step, to more accurately define when the transition takes place.

## 3.3 Fallout and its effect on cloud rise

The way the CRM determines the mass change due to fallout during cloud rise is another area of concern. In the test case as currently modeled in the CRM, fallout's contribution to the buoyancy in the cloud is negligible. However, the documentation states that it can be important, and therefore the CRM spends a fair portion of the computational effort to calculate it during the solving of the ODEs. How it models this was looked at quite closely when trying to emulate the model with Mathematica.

DELFIC treats the cloud as if it were a cylinder when it comes to determining fallout during the rise. For each time step, the CRM determines what distance the particles in the cloud would drop. It then reduces the soil mass by the fraction of the cylinder (equivalently the fraction of the vertical height) which fell out the bottom of the cloud. One must keep in mind that DELFIC assumes the entire mass is present at its initialization time, which is a gross approximation to the actual time dependent loading of the cloud with soil. Therefore allowing for a time dependent soil loss during the rise seems very artificial considering the above assumption. This area will be looked at more closely in the future.

## 3.4 Oscillating cloud height

DELFIC's CRM doesn't allow the cloud center to drop from the maximum height obtained and fixes its height upon reaching this maximum (see lines DERIV 70 to DERIV 80 in Volume II). It was seen using the Mathematica model that the cloud oscillates in altitude before stabilizing at an altitude lower than the maximum, which DELFIC freezes at. McGahan stated that oscillation in cloud height is a physical phenomena observed during the atmospheric tests. This was one of the changes made to the 1979 DELFIC by McGahan which will be retained during the research at AFIT.

## 3.5 Wind shear and its effect on cloud rise

In coding the equations in the higher level language the first time, the effect of wind shear was inadvertently left out. This was due to the fact that it was mentioned as a replacement rule on page 26, Eq. (2.2.13) of Volume I.

$$\frac{S}{V}\mu v \Rightarrow \mu\left(\frac{S}{V}v + \frac{1.5}{R_c}v_s\right) \qquad \text{(EQ 5)}$$

By comparing the results before and after wind shear terms were inserted, it was seen that wind shear has a noticeable effect on the test case (see Figure WIND SHEAR Comparison). DELFIC simply calculates the wind shear, $v_s$, between the velocity vector at the top of the cloud and the bottom. The question remains if the CRM's method of calculating the wind shear is a good approximation considering the vertical extent of some clouds.

## 3.6 Fits to physical data

A question or two came up when modeling the viscosity and specific heat equations used in the CRM. Are they valid for the temperatures experienced in the cloud at early times and are the jumps in the specific heat fits valid? The viscosity equation was found to match existing data up to 2000 K but data for higher temperatures was not available. Although the source of the fits is still not known, it was discovered that the model is very sensitive to the specific heat values. Equations for $\rho$, $P_{ws}$ and P will also be checked because of their importance in determining whether the cloud is saturated and latent heat is released. Therefore this will be an area for further research.

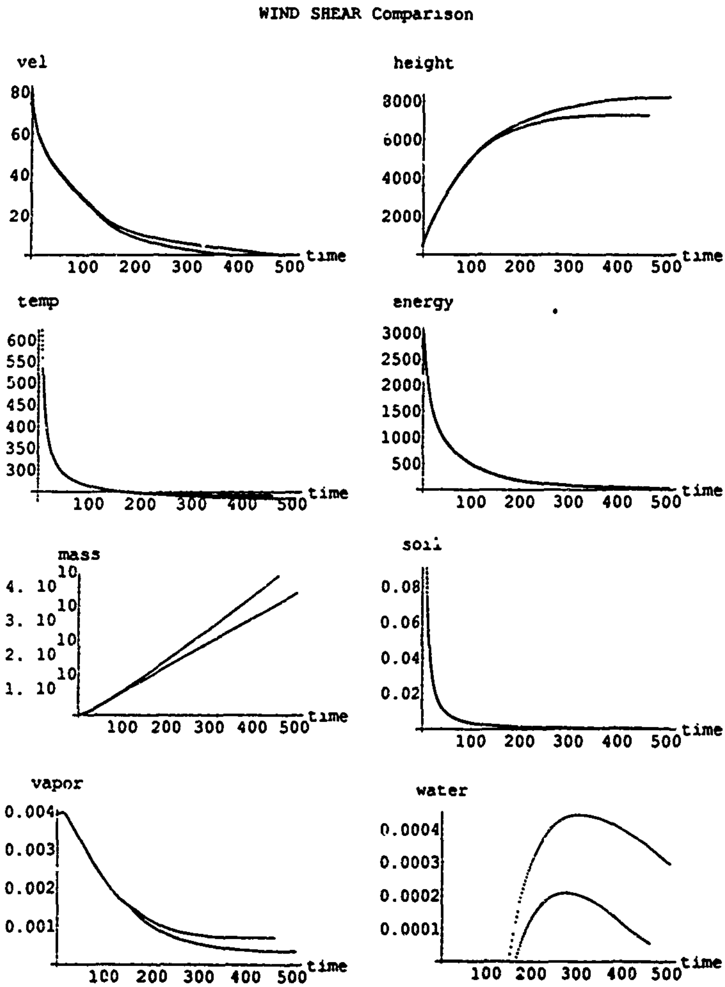
# 4.0 Further Work

The work described in this section are ideas that have come up as to how to improve the CRM of DELFIC. Some of this section represents ideas proposed by other authors. Some of this section is part of the prospectus presented to the author's research committee at AFIT.

## 4.1 Minor's recommendations

It was stated that the assumption made by Norment, that the specific heat of entrained air could be modeled by that of dry air (Norment, 1970), may not be valid. This would require replacing CPAI with a new variable CPI which is the integral of $c_p$ instead of the integral of $c_{pa}$. The code can easily be altered, but the way Minor defines the variable CPI is in

**FIGURE 4. Effect of *removing* wind shear, WIND SHEAR Comparison**



error. The new variable should be CPI=(CPAI+XE*CPWI)/(1.+XE) *not* CPI=(CPAI+X*-CPWI)/(1.+X) as stated (Minor, 1988). This new variable CPI should be used in equations DERIV128 and DERIV113 of the current CRM.

Minor's other area of concern was the modeling of dry and wet fallout separately. He states that P(t) (or F in Volume I) is only soil and not total fallout. He suggests an additional variable, $P_w(t)$, for water fallout. He stated that if P(t) included both dry and wet fallout, the particle size distribution would have depended on water content.

In reviewing his claims, it was found that the original 1967 DELFIC did account for condensation in calculating the density of the particles and their volumes. These changes in particle density and volume were taken out with the first revision of the CRM (Norment, 1970). In addition, the ability to allow condensed water to fall out of the cloud was eliminated in the 1979 version. This will be looked at in conjunction with the bigger question of fallout in general mentioned above.

## 4.2 Heubsch's recommendations

Heubsch's 1975 critique of the revised CRM, stated that the model produced physically unrealistic results or results that disagreed with the experimental data. Upon closer examination of the model, Heubsch proposed the errors were due to the formulation of the equations for cloud mass, velocity, temperature, and dimensions. He found that some of the terms in the equations contained theoretical errors.

Heubsch presented an amended set of cloud equations which removed both the theoretical errors and output discrepancies. He made further recommendations on how to improve the CRM and proposed three levels of effort. His recommendations are important since he was the original author of the CRM. However, since his organization no longer had control of DELFIC, he was not able to mandate implementation of the improvements.

Norment published his own validation study of the revised CRM (Norment, 1977). Norment did change the momentum (or velocity) equation, as Heubsch had recommended, but left the other equations alone. He did adjust some of his empirical parameters, simply as another method of reaching better agreement with experimental data. He pointed out which parameters, using his set of equations, most influenced changes in the results.

Norment did not, however, change the other equations, in particular the entrainment equation. Heubsch recommended keeping only the first term of the entrainment equation based

on current theory. A model which does not contradict current theory *and* matches the test data is needed.

## 4.3 McGahan's recommendations

In addition to the recommendation to go back and correct the violations of theory in the current version of DELFIC's CRM, there exists an outside recommendation. McGahan from SAIC is currently the most knowledgable person with regards to the theory and use of the DELFIC model. He does need someone to review the CRM for its errors, but in addition, he would like to improve the model. He wants to see if it could distribute the particles in a more robust method than the 1D (vertical) placement of particles it now uses. This would then be followed by a comparison with the higher level hydrodynamics codes that are starting to be used for nuclear burst modeling.

His recommendation involves using vortex theory and a two dimensional representation of the cloud particle distribution. In a similar fashion to how DELFIC distributes the particles in 1D after its parcel rise subroutine, vortex theory could be used to provide a 2D flow-field in the rising cloud. By tracing the flow of particles during rise, using the output values from the cloud rise equations, the position, velocity, and acceleration of particles could be tracked up to stabilization.

This would be a new method of calculating the data used by the nuclear effects community which is currently available only from the supercomputer hydrodynamics codes. The goal is to refine the vertical distribution of the larger particles in DELFIC results, which currently stabilize at much lower altitudes than the hydrocodes predict. McGahan thinks that accounting for the spatially varying flow field will allow a better treatment of the large particle tracking, and possibly show them attaining a higher altitude. The implementation of this would also provide a radial distribution of particles that is not currently available from DELFIC. Not only will this tracking give a better definition of the particle distribution with time, but also give dynamic representation of the particles.

## 4.4 DICE/MAZ - TASS results

A hydrodynamic codes are being developed more fully, their output is being trusted more. The idea was suggested that the output of TASS, a 2-D hydrocode, may want to be used to help determine the best values for the remaining parameters in CRM. This would be in addition to the validation efforts with atmospheric test data. This area is still being considered but may not be feasible.

## 5.0 Conclusion

This report showed most of the findings during a recent study of the 1979 version of DELFIC's Cloud Rise Module (CRM). The areas of the code and documentation that still contain errors were listed as well as the suggested corrections. Other areas of possible improvement to the CRM found using a higher level language were discussed. Finally, the proposed work for the current research at AFIT was discussed. These areas include revising the equations in the CRM to be self consistent with conservation laws and available data. Also the inclusion of a 2-D vortex flow field for particle placement within the cloud will be accomplished.

# List Of References


Heubsch, 1967, The Department of Defense Land Fallout Prediction System. Volume III. Cloud Rise. (No longer in DTIC - available through DASIAC loan).

Heubsch, 1975, Analysis and Revision of the Cloud Rise Module of the Department of Defense Land Fallout Prediction System (DELFIC). ADB 007-607.

Minor, 1988, The Effect of Water Content on the Predictions of the Cloud Rise Module of DELFIC. ADA 194-618.

Norment, 1970, The Department of Defense Land Fallout Prediction System. Volume III. Cloud Rise. Revised. AD 879-890.

Norment, 1977, Validation and Refinement of the DELFIC Cloud Rise Module. ADA 047-372.

Norment, 1979a, DELFIC: Department of Defense Fallout Prediction System. Volume I - Fundamentals. ADA 088-367.

Norment, 1979b, DELFIC: Department of Defense Fallout Prediction System. Volume II - User's Manual. ADA 088-512.

# APPENDIX A

## Cloud Equations

DRY EQUATIONS

$$\frac{du}{dt} = \left(\frac{T^*}{T_e^*}\beta' - 1\right) g - \left(\frac{2k_2 v}{H_c}\frac{T^*}{T_e^*}\beta' + \frac{1}{m}\frac{dm}{dt}\right) u$$

$$\frac{dT}{dt} = -\frac{\beta'}{\bar{c}_p(T)}\left[\frac{T^*}{T_e^*} g u + \left(\int_{T_e}^{T} c_{pa}(T)\, dT\right)\frac{1}{\beta' m}\frac{dm}{dt}\bigg|_{ent} - \varepsilon\right]$$

$$\frac{dE}{dt} = 2k_2\frac{T^*}{T_e^*}\beta'\frac{u^2 v}{H_c} + \frac{u^2}{2}\frac{1}{m}\frac{dm}{dt}\bigg|_{ent} - E\frac{1}{m}\frac{dm}{dt}\bigg|_{ent} - \varepsilon$$

$$\frac{dm}{dt}\bigg|_{ent} = \frac{\beta' m}{1 - \frac{\beta'}{T^*\bar{c}_p}\int_{T_e}^{T} c_{pa}(T)\, dT}\left(\frac{S}{V}\mu v + \frac{\beta'}{T^*\bar{c}_p}\left[\frac{T^*}{T_e^*} g u - \varepsilon\right] - \frac{g u}{R_a T_e^*}\right)$$

$$\frac{ds}{dt} = -\frac{1}{\beta'}\frac{1+x}{1+x_e} s \frac{1}{m}\frac{dm}{dt}\bigg|_{ent} - \frac{1+x+s+w}{m}\left(\frac{s}{s+w}\right) F$$

$$\frac{dx}{dt} = -\frac{1+x+s}{1+x_e}(x - x_e)\frac{1}{m}\frac{dm}{dt}\bigg|_{ent}$$

$$\frac{dw}{dt} = -\frac{1}{\beta'}\left(\frac{1+x}{1+x_e}\right)(w + x - x_e)\frac{1}{m}\frac{dm}{dt}\bigg|_{ent} - \frac{dx}{dt}$$

$$\varepsilon = \frac{k_3 (2E)^{3/2}}{H_c}$$

$$v = Max(u, \sqrt{2E})$$

WET EQUATIONS:

$$\left.\frac{dm}{dt}\right|_{ent} = \frac{\beta' m}{1 - \frac{1}{T^*}\left[\frac{\beta'}{1 + \frac{L^2 x\xi}{c_p R_a T^2}}\right]\left[T - T_e + \frac{L(x - x_e)}{c_p}\right]}\left\{\frac{S}{V}\mu v + \left[\frac{\beta'/T^*}{1 + \frac{L^2 x\xi}{c_p R_a T^2}}\right]\left[\frac{guT^*}{T_e^* c_p}(1 + \frac{Lx}{R_a T}) - \frac{\varepsilon}{c_p}\right] - \frac{gu}{R_a T_e^*}\right\}$$

$$\frac{dT}{dt} = -\frac{\beta'}{1 + \frac{L^2 x\xi}{c_p R_a T^2}}\left[\left((T - T_e) + \frac{L(x - x_e)}{c_p}\right)\frac{1}{m\beta'}\left.\frac{dm}{dt}\right|_{ent} + \frac{T^*}{T_e^*}\frac{g}{c_p}u(1 + \frac{Lx}{R_a T}) - \frac{\varepsilon}{c_p}\right]$$

$$\frac{1}{x}\frac{dx}{dt} = (1 + x/\xi)\frac{L\xi}{R_a T^2}\frac{dT}{dt} + (1 + x/\xi)\frac{g}{R_a T_e^*}u$$

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188



1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
May 1993

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Critique of DELFIC's Cloud Rise Module

5. FUNDING NUMBERS

6. AUTHOR(S)
Vincent J. Jodoin, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Engineering Physics
Air Force Institute of Technology
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/EN/TR/93-04

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a DISTRIBUTION/AVAILABILITY STATEMENT
Approval for public release;
Distribution unlimited

12b DISTRIBUTION CODE

13 ABSTRACT (*Maximum 200 words*)

This report discusses the Cloud Rise Module (CRM) of the Defense Land Fallout Interpretive Code (DELFIC), the DOD's reference fallout code. The first section discusses the errors found in the 1979 documentation. These errors have been corrected and further research by the Air Force Institute of Technology will include the corrections listed in this section. The second section will discuss the use of Mathematica in simulating the CRM. This part of the report will describe how a higher level language was used to review the results of DELFIC's CRM and uncover some problem areas. The final part of this report will describe further work as planned by the author in improving the CRM of DELFIC.

14. SUBJECT TERMS
DELFIC, Nuclear Clouds, Fallout, Cloud Rise

15. NUMBER OF PAGES
23

16. PRICE CODE

| 17 SECURITY CLASSIFICATION OF REPORT | 18 SECURITY CLASSIFICATION OF THIS PAGE | 19 SECURITY CLASSIFICATION OF ABSTRACT | 20 LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UL |